DETAILED ACTION

1. The Office has withdrawn all rejections raised in the Final Rejection mailed 10/18/2007.

Examiner's Amendment

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with David Rouille, Reg. No. 40,150, on 4/22/2008, and Christopher Lutz, Reg. No. 44,883 on 4/22 and 4/23/2008.

The application has been amended as follows:

In the claims:

3. Amend claims 1, 18, 21, 38, 39 and 42, as follows:

1. (Currently Amended) An encoded set of processor based instructions on a computer readable storage medium that, when executed in a computer having the processor, cause the computer to perform a method of monitoring access to a protected database resource comprising:

identifying an attempt to access the database resource, the access attempt being local and directed to an access gateway of the database resource;

identifying a plurality of access paths to the protected database resource;

intercepting the identified attempt to access the database resource, intercepting occurring in a prioritized manner with respect to receipt of the access attempt by the access gateway, intercepting further comprising:

determining an interprocess communication (IPC) mechanism to be employed by a local client for accessing the database resource, the IPC mechanism defined by a dynamic linked library (DLL);

identifying, via a common access point for the access paths to the protected database resource, access attempts occurring via identified access point for the identified access paths, identifying the access attempts including:

replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying a local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as an entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway; and

establishing, in response to an identified connection attempt, an event notification list responsive to database access attempts using an identified connection;

establishing an IPC intercept from the common access point employed by database clients for accessing the database resource by storing, in the event notification list, the local agent as first entity to receive control from a database access attempt via the identified connection, the event notification list operable to initiate handlers responsive to event;

receiving the access attempt at the local agent via the IPC intercept prior to receipt of the access attempt by the access gateway; and

transmitting, in a nondestructive manner, the intercepted access attempt to the local agent, the nondestructive manner operable to preserve the intercepted access attempt for successive receipt by the access gateway.

18. (Currently Amended) An encoded set of processor based instructions on a computer readable storage medium that, when executed in a computer having the processor, cause the computer to perform a method for controlling local access to a database comprising:

identifying a local access gateway to the database, the access gateway being a common access point into the database;

establishing an interception wrapper between a local client and the access gateway, establishing the interception wrapper further comprising:

identifying, at least one interprocess communication (IPC) operation, each of the identified IPC operation corresponding to an event, the event derived from a database instruction, the IPC operation defined by a dynamic linked library (DLL);

replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying a local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as an entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway; and

establishing, in response to identified connection attempt, an event notification list responsive to database access attempts using identified connection;

instantiating a local event object corresponding to the event, the local event object having a notification list indicative of notifications of an object to be made upon an occurrence of the event; and

storing, in the event notification list, the local agent as first entity to receive control from a database access attempt via the identified connection, the event notification list operable to initiate handlers responsive to the event;

intercepting, via the interception wrapper, an access attempt from a local client prior to receipt of the access attempt by the access gateway, the access attempt indicative of a pending database instruction in an IPC buffer;

identifying the local event object corresponding to the access attempt;

indexing the notification list corresponding to the identified local event object;

traversing the indexed notification list, the notification list including entries of notifications to be performed upon occurrence of the event;

reading a traversed entry corresponding to the local agent, the entry indicative of the location of the local agent;

notifying the local agent using the read location of the local agent;

retrieving, in response to the notification, the database instruction from the IPC buffer;

transmitting the retrieved database instruction from the IPC buffer to a data security device operable to analyze the propriety of the database instruction;

reading a successive traversed entry corresponding to the access gateway, the entry indicative of the location of the access gateway; and

notifying, after the notifying of the local agent, the access gateway of the IPC event occurrence using the read location of the access gateway.

21. (Currently Amended) A local agent ~~comprising~~ stored on a computer readable storage medium having an encoded set of processor based instructions that, when executed by a processor responsive to the instructions, performs steps for monitoring access to a protected database resource comprising:

an interface operable to identify an attempt to access the database resource, the access attempt being local and directed to an access gateway of the database resource, the access attempt being deterministic of a database instruction, a local agent being in communication with a data security device operable to analyze the propriety of the access attempt from objects and data values referenced by the DB instruction;

an interprocess communication IPC intercept operable to intercept the identified attempt to access the database resource, the IPC intercept defined by a dynamic linked library (DLL),

identifying the attempt including replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying the local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as an entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway;

intercepting occurring in a prioritized manner with respect to receipt of the access attempt by the access gateway, the prioritized manner including:

establishing, in response to an identified connection attempt, an event notification list responsive to database access attempts using an identified connection; and

storing, in the event notification list, the local agent as the first entity to receive control from a database access attempt via the identified connection, the event notification list operable to initiate handlers responsive to an event;

the local agent further operable to transmit, in a nondestructive manner, the intercepted access attempt to a data security device, the nondestructive manner operable to preserve the intercepted access attempt for successive receipt by the access gateway, the local agent further operable to reroute the intercepted access attempts to the data security device, the data security device operable to offload data security decisions as a consolidated appliance, the offloaded data security decisions relieving a host from processing the data security decisions.

38. (Currently Amended) A data security device for monitoring access to a protected database resource comprising:

a memory comprising a computer readable storage medium operable to store an encoded set of processor based instructions performable by a local agent;

a processor operable to execute the instructions in the memory;

an interface operable for interconnection with a database host, the data security device in communication with the local agent on the database host, the local agent responsive to the instructions to:

identify an attempt to access the database resource, the access attempt being local and directed to an access gateway of the database resource via an interprocess communication (IPC) mechanism defined by a dynamic linked library (DLL);

intercept the identified attempt to access the database resource, intercepting occurring in a prioritized manner with respect to receipt of the access attempt by the access gateway, intercepting further comprising:

replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying the local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as an entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway;

identifying, at least one interprocess communication operation, each of the identified IPC operation corresponding to an event, the event derived from a database instruction;

establishing, in response to an identified connection attempt, an event notification list responsive to database access attempts using an identified connection;

instantiating a local event object corresponding to the event, the local event object having a notification list indicative of notifications of an object to be made upon an occurrence of the event; and

storing, in the event notification list, the local agent as an first entity to receive control from a database access attempt via the identified connection, the event notification list operable to initiate handlers responsive to the event; and

transmit, in a nondestructive manner, the intercepted access attempt to a local agent, the nondestructive manner operable to preserve the intercepted access attempt for successive receipt by the access gateway.

39. (Currently Amended) A computer program product having a computer readable storage medium operable to store computer program logic embodied in computer program code encoded as a set or processor based instructions thereon for, when executed by a processor in a computer, perform steps for monitoring access to a protected database resource comprising:

computer program code for identifying an attempt to access the database resource, the access attempt being local and directed to an access gateway of the database resource;

computer program code for intercepting the identified attempt to access the database resource, intercepting occurring in a prioritized manner with respect to receipt of the access attempt by the access gateway, computer program code for intercepting further comprising:

computer program code for determining an interprocess communication (IPC) mechanism to be employed by a local client for accessing the database resource, the IPC mechanism defined by a dynamic linked library (DLL);

computer program code for identifying, via a common access point for the access paths to the protected resource, access attempts occurring via the identified access point for the identified access paths, identifying the access attempts including:

replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying the local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as the entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway; and

establishing, in response to an identified connection attempt, an event notification list responsive to database access attempts using an identified connection;

computer program code for establishing an IPC intercept from the common access point employed by database clients for accessing the DB resource by storing, in the event notification list, a local agent as the first entity to receive control from a database access attempt via the identified connection, the event notification list operable to initiate handlers responsive to an event; and

receiving the access attempt at the local agent via the IPC intercept prior to receipt of the access attempt by the access gateway; and

computer program code for transmitting, in a nondestructive manner, the intercepted access attempt to the local agent, the nondestructive manner operable to preserve the intercepted access attempt for successive receipt by the access gateway.

42. (Currently Amended) An encoded set of processor based instructions on a computer readable storage medium that, when executed by a processor responsive to the instructions, perform a method of monitoring access to a protected database resource comprising:
identifying an attempt to access the database resource, the access attempt being local and directed to an access gateway of the database resource, identifying the access attempt further comprising listening, at a common access point, for an incoming connection to the database resource, the common access point adapted to aggregate access attempts to the database resource from a plurality of access mediums;
intercepting the identified attempt to access the database resource, intercepting occurring in a prioritized manner with respect to receipt of the access attempt by the access gateway, intercepting further comprising:

determining an interprocess communication (IPC) mechanism to be employed by a local client for accessing the database resource, the IPC mechanism defined by a dynamic linked library (DLL);

identifying, via a common access point for the access paths to the protected database resource, access attempts occurring via an identified access point for an identified access paths, identifying the access attempts including:

replacing the ~~determined~~ defined DLL for database access with an interface wrapper, the interface wrapper for identifying a local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as an entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway; and

establishing, in response to an identified connection attempt, an event notification list responsive to database access attempts using an identified connection, the access attempt being deterministic of a database instruction, such that the local agent is in communication with a data security device operable to analyze a propriety of the access attempt from objects and data values referenced by the database instruction;

establishing an IPC intercept from the common access point employed by database clients for accessing the database resource by storing, in the event notification list, a local agent as a first entity to receive control from a database access attempt via an identified connection, the event notification list operable to initiate handlers responsive to an event; and

intercepting the access attempt at the local agent via the IPC intercept prior to receipt of the access attempt by the access gateway; and

receiving, in a nondestructive manner, the intercepted access attempt by a local agent, the nondestructive manner operable to preserve the intercepted access attempt for successive receipt by the access gateway, transmitting further comprising rerouting the intercepted access attempts to the data security device, the data security device operable to offload data security decisions as a consolidated appliance, the offloaded data security decisions relieving the host from processing the data security decisions.

4. After a thorough search, and in light of the prior art of record, claims 1-2, 6, 3-8, 11-21, 23-24, 37, 25, 27-36, 38-39, and 42-45 (renumbered as 1-39) are allowed.

Reasons For Allowance

5. This section sets forth an examiner's statement of reasons for allowance and is structured as follows: A) A description of Applicant's subject matter; B) A description of the prior art used in the last rejection; and C) The differences between the claimed subject matter and the cited prior art of the last rejection.

A. The present invention is directed to an intrusion detection analysis system for the analysis of database accesses.

B. The closest prior art, Krack et al (US Patent No. 6,941,369), is directed to techniques for ensuring secure access to data resources via a CGI gateway. The further cited reference, Jai Sundar Balasubramaniyan et al.,("An Architecture for Intrusion Detection Using Autonomous Agents", 14th Annual Computer Security Applications Conf. Proc., Phoenix, AZ, Dec. 7-11, 1998, pp. 13-24), is directed to the techniques for intrusion detection employing a collection of independent agents. The additionally cited reference, Muralidaran Gangadharan et al. ("Intranet Security with Micro-Firewalls and Mobile Agents for Proactive Intrusion Response", IEEE Int'l Conf. on Computer Networks and Mobile Computing, Beijing, China, Oct. 16-19, 2001, pp. 325-

332), is directed to techniques for intrusion detection on enterprise Intranets using micro-firewalls and mobile agents.

C. These references do not disclose the use of an interface wrapper for identifying a local agent as a responsive entity for database calls such that the interface wrapper spoofs the database access gateway as the entity responsive to database access attempts, the interface wrapper thus defining an IPC intercept for receiving database connection attempts prior to receipt by the database access gateway and establishing, in response to the identified connection attempt, an event notification list responsive to database access attempts using the identified connection.

6. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Non-Patent Literature

Chari, Suresh N., et al., "BlueBoX: A Policy-Driven, Host-Based Intrusion Detection System", ACM Transactions on Information and System Security, Vol. 6, No. 2, May 2003, pp. 173-200.

Schepers, Filip, et al., "Network- Versus Host-Based Intrusion Detection", Information Security Technical Report, Vol. 3, Issue 4, © 1998, pp. 32-42.

Levine, John, et al., "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks", Proc. of the 2003 IEEE Workshop on Information Assurance, West Point, NY, Jun. 18-20, 2003, pp. 92-99.

Kewley, Dorene L., et al., "DARPA Information Assurance Program Dynamic Defense Experiment Summary", IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, Vol. 31, No. 4, Jul. 2001, pp. 331-336.

US Patent Application Publications

Gordy et al	2005/0005031
Royer	2002/0157020

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert Stevens whose telephone number is (571) 272-4102. The examiner can normally be reached on M-F 6:00 - 2:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Cam Y Truong/
Primary Examiner, Art Unit 2162

/Robert Stevens/
Examiner
Art Unit 2162

April 24, 2008